R_Project_NT

Nelson Tran

2023-07-31

```
library(dplyr)
```

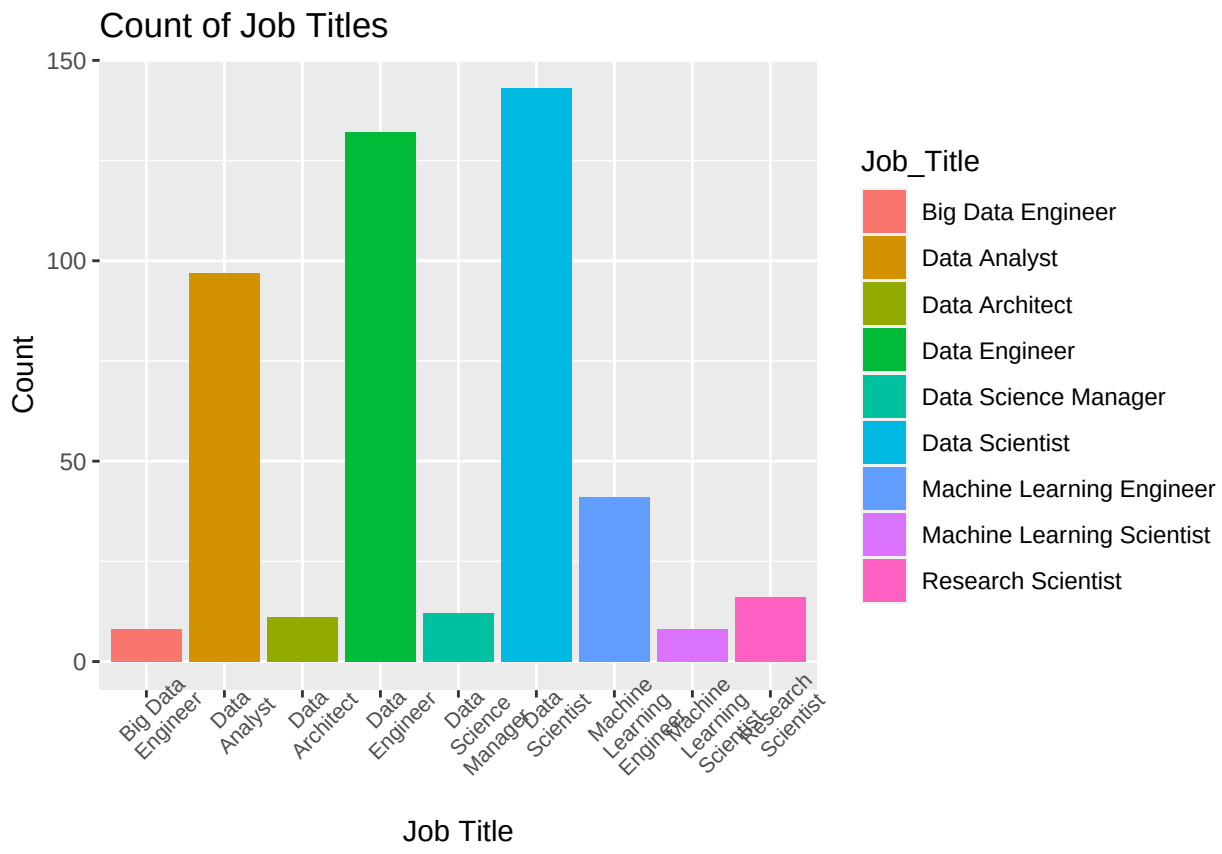
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
#Reading sales.csv

job_data <- read.csv("data.csv",
                 stringsAsFactors = FALSE,
                 )
```

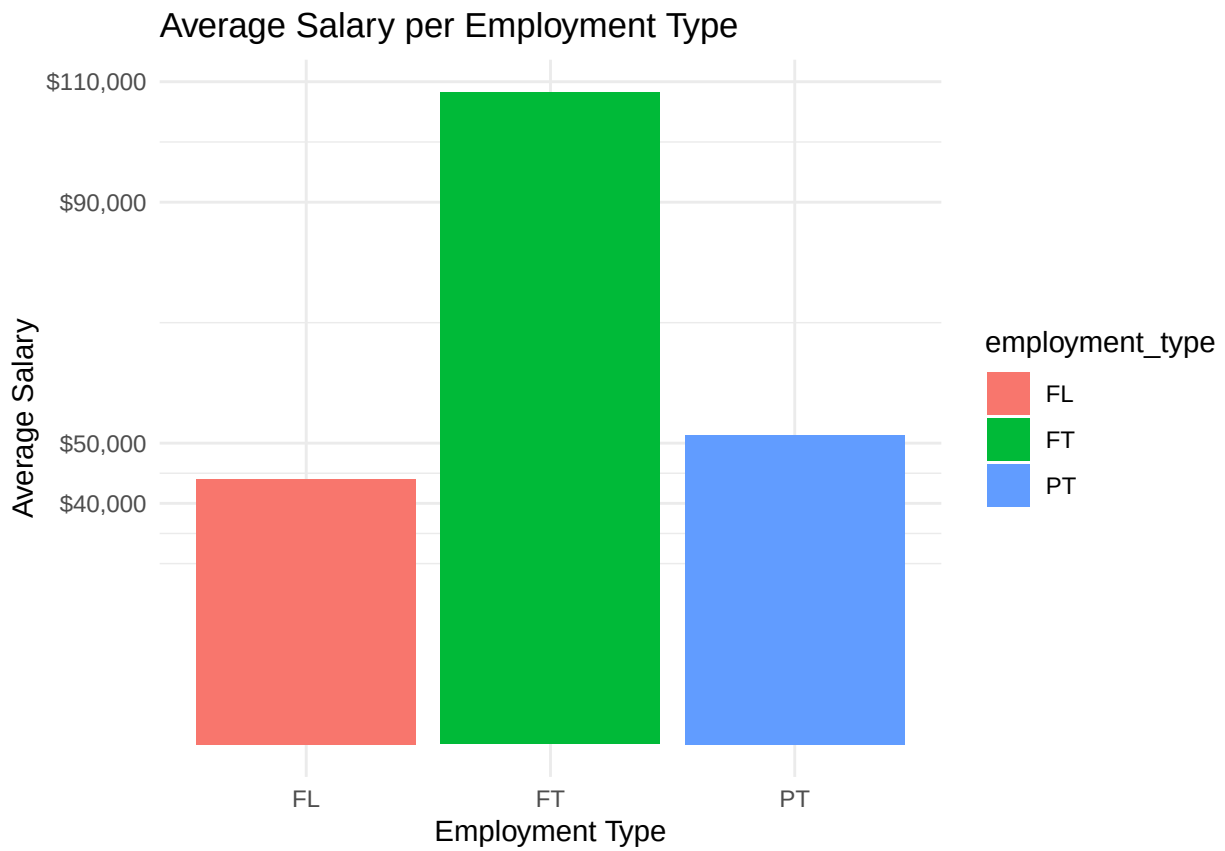
```
#data cleaning

title_counts <- table(job_data$job_title)
title_counts <- as.data.frame(title_counts)
colnames(title_counts) <- c("Job_Title", "Count")

#Used a threshold of 7 counts of a job title to help keep data from skewing salary level
threshold <- 7
title_counts <- subset(title_counts, Count > threshold)

ggplot(title_counts, aes(x = Job_Title, y = Count, fill = Job_Title)) +
  geom_bar(stat = "identity") +
  labs(title = "Count of Job Titles", x = "Job Title", y = "Count") +
  scale_x_discrete(labels=scales::label_wrap(10)) +
  theme(axis.text.x = element_text(angle = 45, size = 8))
```



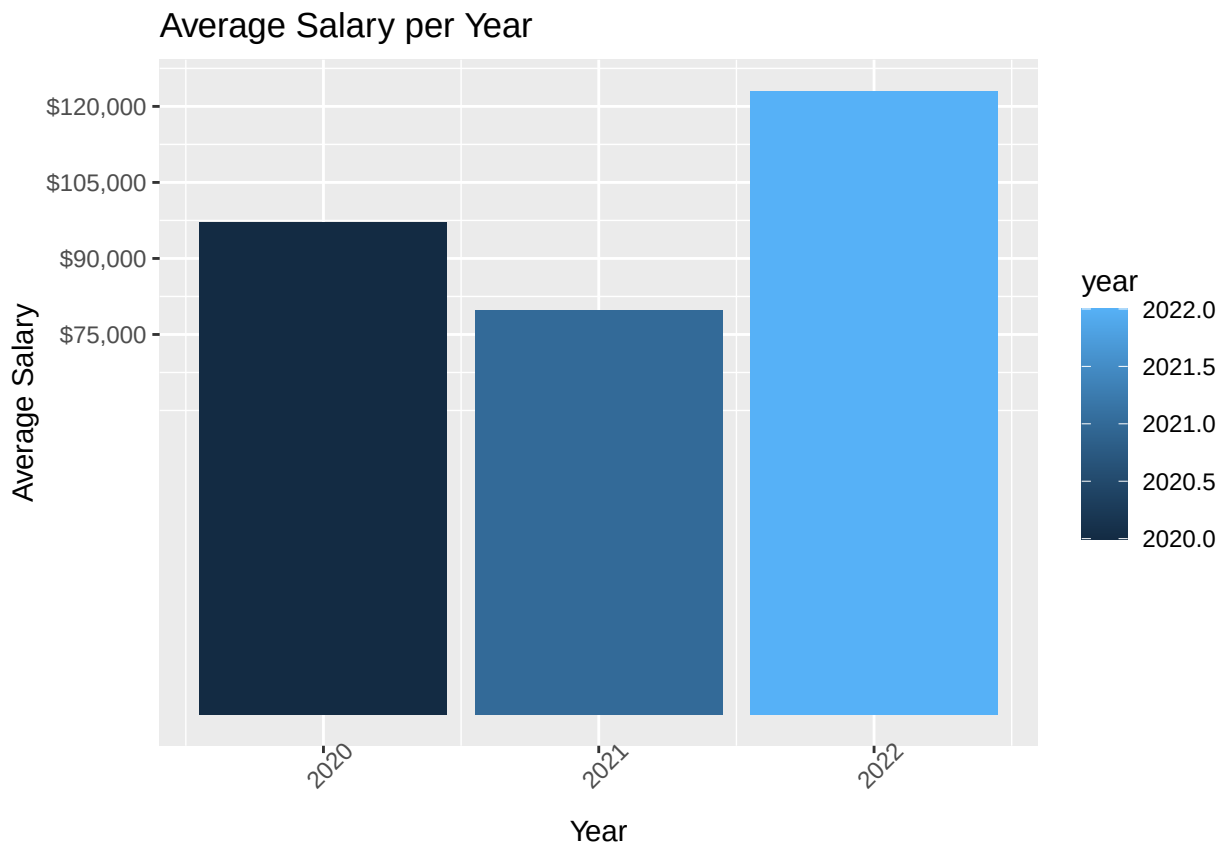

```
#removing least common job titles

Job_titles_to_keep <- c("Big Data Engineer", "Data Analyst", "Data Artchitect",
                        "Data Engineer", "Data Science Manager", "Data Scientist",
                        "Machine Learning Engineer", "Machine Learning Scientist",
                        "Research Scientist")

filtered_job_data <- filter(job_data, job_title %in% Job_titles_to_keep)
```

```
#Average Salary for Experience Level

avg_salary_exp <- aggregate(filtered_job_data$salary_in_usd,
                            list(filtered_job_data$experience_level),
                            FUN=mean)

colnames(avg_salary_exp) <- c("experience_level", "average_salary")

ggplot(avg_salary_exp, aes(x=experience_level, y=average_salary, fill=experience_level)) +
  geom_bar(stat = "identity") +
  labs(title="Average Salary for Experience Level", x='Experience_level', y='Average Salary') +
  theme_minimal() +
  scale_y_continuous(labels=scales::dollar_format(), breaks = c(60000,80000,130000,200000))
```

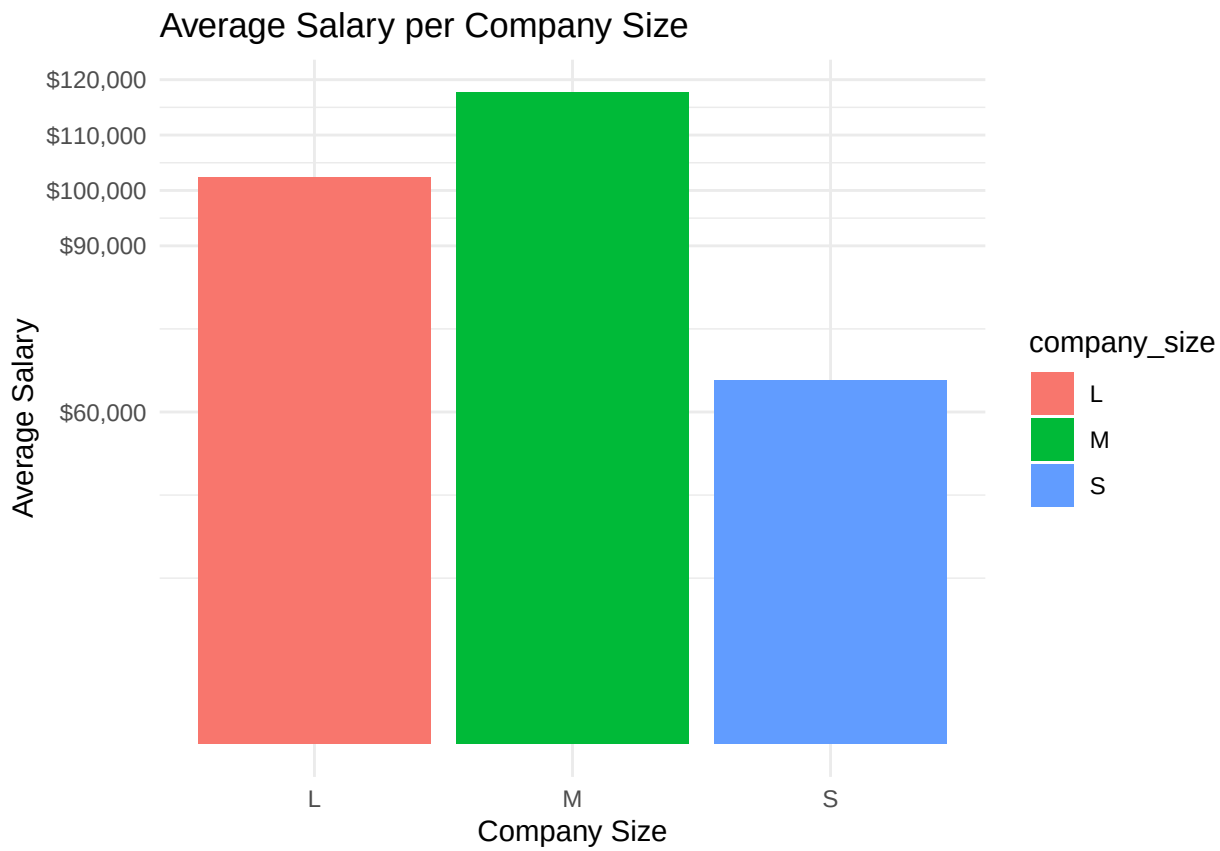

```
#Avg Salary of Employment type

avg_salary_type <- aggregate(filtered_job_data$salary_in_usd,
                             list(filtered_job_data$employment_type),
                             FUN=mean)

colnames(avg_salary_type) <- c("employment_type", "average_salary")

ggplot(avg_salary_type, aes(x=employment_type, y=average_salary, fill=employment_type)) +
  geom_bar(stat = "identity") +
  labs(title = 'Average Salary per Employment Type', x = 'Employment Type', y = "Average Salary") +
  theme_minimal() +
  scale_y_continuous(labels=scales::dollar_format(), breaks = c(40000,50000,90000,110000))
```




```
#Average salary per year in USD

avg_salary_year <- aggregate(filtered_job_data$salary_in_usd,
                             list(filtered_job_data$work_year),
                             FUN=mean)

colnames(avg_salary_year) <- c("year", "average_salary")

ggplot(avg_salary_year, aes(x = year, y = average_salary, fill = year)) +
  geom_bar(stat = "identity") +
  labs(title = 'Average Salary per Year', x = 'Year', y = 'Average Salary') +
  theme(axis.text.x = element_text(angle=45)) +
  scale_y_continuous(breaks = c(75000,90000,105000,120000), labels=scales::dollar_format())
```




```
#average salary for company size

avg_salary_comp_size <-aggregate(filtered_job_data$salary_in_usd,
                                 list(filtered_job_data$company_size), FUN=mean)

colnames(avg_salary_comp_size) <- c("company_size", "average_salary")

ggplot(avg_salary_comp_size, aes(x=company_size, y=average_salary, fill=company_size)) +
  geom_bar(stat = "identity") +
  labs(title = 'Average Salary per Company Size', x = 'Company Size', y = 'Average Salary') +
  theme_minimal() +
  scale_y_continuous(breaks = c(60000,90000,100000,110000,120000), labels=scales::dollar_format())
```



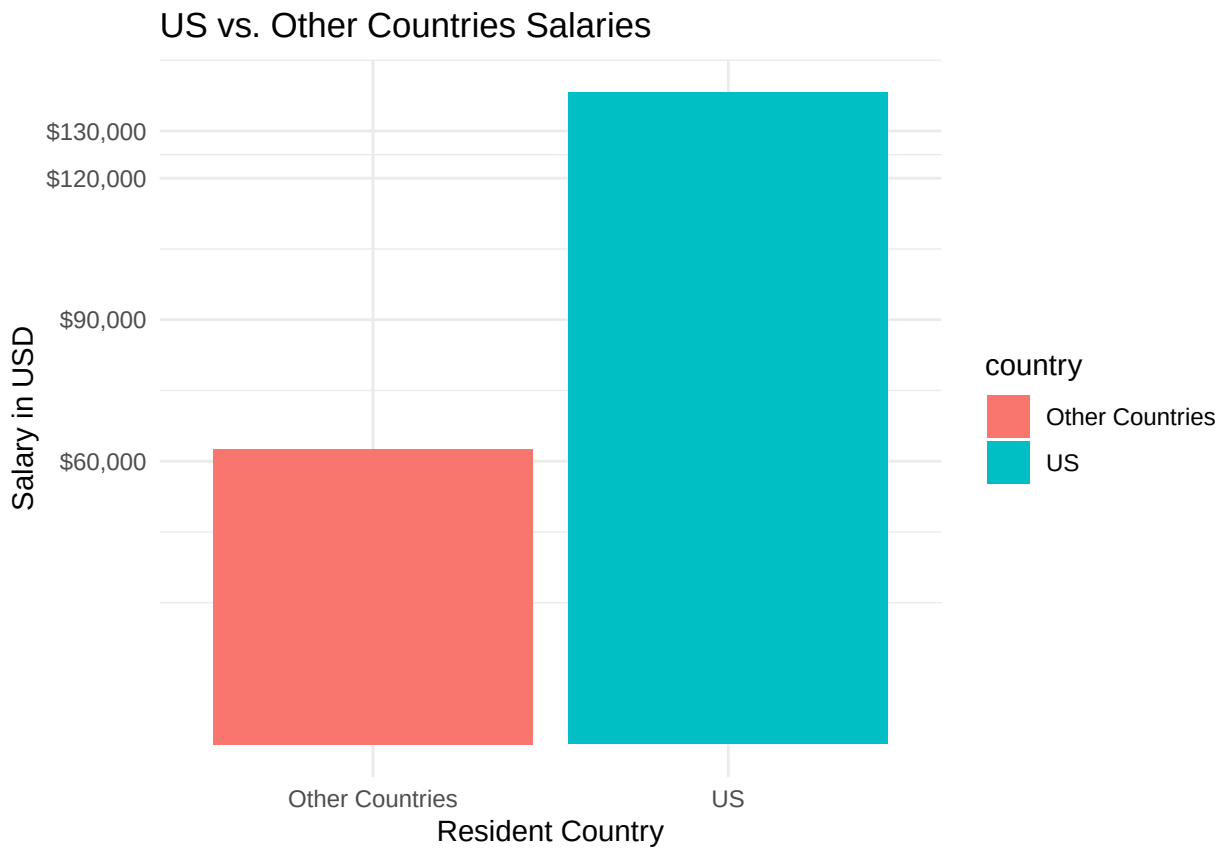

```
salary_us <- filtered_job_data[filtered_job_data$employee_residence
                               == "US", "salary_in_usd"]
salary_other <-filtered_job_data[filtered_job_data$employee_residence
                               != "US", "salary_in_usd"]
salary_other <-as.data.frame(salary_other)
salary_us <- as.data.frame(salary_us)

mean_salary_us <- mean(salary_us$salary_us, na.rm = TRUE)
print(mean_salary_us)
```

```
## [1] 139676.3
```

```
mean_salary_other <- mean(salary_other$salary_other, na.rm = TRUE)
print(mean_salary_other)
```

```
## [1] 61961.97
```

```
compared_salary <- data.frame(country = c('US', 'Other Countries'),
                              mean_salary = c(138199.5, 62546.53))

ggplot(compared_salary, aes(x = country, y = mean_salary, fill = country)) +
  geom_bar(stat = 'identity') +
  labs(title = 'US vs. Other Countries Salaries', x = 'Resident Country',
       y = 'Salary in USD') +
  theme_minimal() +
  scale_y_continuous(labels = scales::dollar_format(), breaks =
                       c(60000, 90000, 120000, 130000))
```




```
#Taking out part-time and freelance workers since CEO would most likely hire a full time employee
Employment_type_to_keep <- c("FT")

filtered_job_data <- filter(filtered_job_data, employment_type %in% Employment_type_to_keep)

#Filtering out executive level job salaries since that is an outlier in this data
length(grep("EX", filtered_job_data$experience_level))
```

```
## [1] 6
```

```
exp_level_tp_keep <- c("EN", "MI", "SE")

filtered_job_data <- filter(filtered_job_data, experience_level %in% exp_level_tp_keep)
```

```
#Suggested Salary for Data Science Hire

ggplot(filtered_job_data, aes(x=work_year, y=salary_in_usd)) +
  geom_point() +
  geom_smooth(method = "lm", aes(y=salary_in_usd, x=work_year)) +
  labs(title = "Suggested Salary for Hire", x= "Year", y="Salary") +
  facet_grid (~experience_level) +
  scale_y_continuous(labels = scales::dollar_format())+
  theme(axis.text.x = element_text(angle=45))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


Suggested Salary for Hire
EN
MI
SE
$400,000
$300,000
$200,000
$100,000
$0
Salary
2020.0
2020.5
2021.0
2021.5
2022.0
Year